# Reliable Computing in the Dark/Dim Silicon Era

Siddharth Garg
New York University
sg175@nyu.edu

## Abstract

Although technology scaling enables the integration of more transistors on a chip, it has been observed that the power consumption per transistor has, in the recent past, not been scaling proportionally with transistor area. As a consequence, given a fixed power budget, not all transistors on the chip can be powered on simultaneously, resulting in the so-called "dark silicon" problem [1]. The parts of the chip that are not powered on are referred to as "dark." An alternative view of a dark silicon chip is to think of it as *overprovisioned* with more transistors than the power budget would mandate and in this context, the question that arises is whether overprovisioning the chip provides any advantages that make-up for the design and manufacturing cost of doing so.

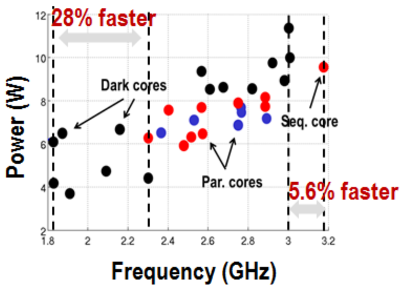

**Figure 1** *Scatter plot of power and frequency for 32 cores on a chip, and which cores are dark (black) and powered on (red,blue).*

The traditional view is that chips can be overprovisioned with heterogeneous computing resources (different types of processing cores, application-specific accelerators, etc.) and the on-chip diversity can be exploited at run-time depending on the application needs. A second view is that even a chip with overprovisioned homogeneous cores (i.e., all processing cores on the chip are micro-architecturally identical) can be leveraged in the dark silicon era. This is for several reasons:

- *Functional yield*: faulty cores can be kept dark without compromising performance.
- *Parametric yield*: core-to-core variations in power and performance due to manufacturing variations can be exploited to maximize parametric yield.
- *Near-threshold operation:* the overprovisioned cores can be powered on at lower voltage and frequency levels, providing greater parallelism.

In this talk, I will discuss the parametric yield and near-threshold voltage aspects, focusing particularly on the implications for predictable and reliable computing.

*Addressing Parametric Yield Using Cherry-picking.* Due to the impact of manufacturing process variations, even identical cores on a chip can have very different power consumption and maximum operating frequency values. For an over-provisioned homogeneous chip multi-processor, one can then pick the optimum subset of cores that maximize performance within a power budget [2]. In particular, for a multi-threaded application, the sequential components can be mapped to the fastest core, while the set of most power-efficient cores can be used for the parallel components. This can be observed in Figure 1, where we note that the sequential core on a 32-core chip is 5.2% faster than the one on a 16-core chip, since there are 16 extra cores to choose from.

From a compiler perspective, there might be opportunities to expose the process variation information to the compiler, allowing it to create low and high power/performance versions of each thread, with the former being mapped to high power cores, while the latter being mapped to lower power cores.

*Near-threshold Computing and Soft Errors.* It might seem, on first glance, a tempting proposition to power on a potentially large number of cores at near-threshold voltage and frequency. For highly parallel applications, it might indeed be beneficial from a performance perspective to execute many threads, albeit at lower frequency values. However, we have recently observed that this comes with a potentially significant cost – increased soft-error rates [3]. This is because in near-threshold mode, more silicon area is exposed to soft-errors, and at the same time, because of low-voltage operation, transistors are more susceptible to particle strikes. Our preliminary results indicate up to an order of magnitude greater likelihood of faults due to soft-errors in the near-threshold mode.

As before, the compiler can use techniques like instruction duplication and control-flow checking when executing in near-threshold mode to mitigate the soft-error issue, but this would come at significant performance cost.

## References

[1] Esmaeilzadeh, Hadi, et al. "Dark silicon and the end of multicore scaling." Computer Architecture (ISCA), 2011 38th Annual International Symposium on. IEEE, 2011.

[2] Raghunathan, Bharathwaj, et al. "Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors." Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, 2013.

[3] Shafique, Muhammad, et al. "The EDA Challenges in the Dark Silicon Era: Temperature, Reliability, and Variability Perspectives." Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference. ACM, 2014